

Claims

We claim:

1. A computer-implemented method for programmatically generating a new graphical program, comprising:
5 executing a graphical program generation (GPG) program;
the GPG program receiving information, wherein the information specifies functionality of the new graphical program;
the GPG program programmatically generating the new graphical program in
10 response to said information specifying the functionality of the new graphical program, wherein the new graphical program implements the specified functionality.
2. The method of claim 1, wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying
15 the new graphical program during said creating.
3. The method of claim 1, wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the new graphical program.
20
4. The method of claim 1, wherein the new graphical program comprises a block diagram portion comprising a plurality of interconnected nodes and a user interface portion;
wherein said programmatically generating the new graphical program includes
25 generating the block diagram portion and the user interface portion.
5. The method of claim 1, wherein said programmatically generating the new graphical program comprises:

creating a plurality of graphical program objects in the new graphical program;
and

interconnecting the plurality of graphical program objects in the new graphical program;

5 wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

6. The method of claim 1, wherein said programmatically generating the new graphical program comprises:

10 creating one or more user interface objects in the new graphical program, wherein the one or more user interface objects perform one or more of providing input to or displaying output from the new graphical program.

15 7. The method of claim 1,
wherein the new graphical program is a virtual instrument.

8. The method of claim 1,
wherein the GPG program is a graphical program.

20 9. The method of claim 1,
wherein the information received by the GPG program specifies a computational process;

wherein the GPG program is operable to generate a new graphical program that implements the specified computational process.

25

10. The method of claim 1,
wherein the information received by the GPG program specifies an algorithm;
wherein the GPG program is operable to generate a new graphical program that implements the specified algorithm.

wherein the information received by the GPG program specifies an instrumentation function;

wherein the GPG program is operable to generate a new graphical program that implements the specified instrumentation function.

5

17. The method of claim 16,

wherein the instrumentation function comprises one or more of:

a test and measurement function; or

an industrial automation function.

10

18. The method of claim 1,

wherein the information received by the GPG program comprises information regarding an existing program having program functionality;

wherein the GPG program is operable to generate a new graphical program that implements at least a portion of the program functionality of the existing program.

15

19. The method of claim 18,

wherein the existing program is a graphical program.

20

20. The method of claim 1,

wherein the GPG program is operable to generate a plurality of new graphical programs, depending on the received information.

21. The method of claim 1,

wherein the new graphical program generated by the GPG program has program functionality;

25

wherein the GPG program is operable to determine at least a portion of the program functionality independently of the received information.

22. The method of claim 1,

wherein the GPG program is operable to generate the new graphical program such that the new graphical program implements additional functionality in addition to the functionality specified by the received information.

5

23. The method of claim 1,

wherein the new graphical program comprises graphical program code;

wherein the GPG program is operable to receive code generation information specifying how to generate at least a portion of the graphical program code.

10

24. The method of claim 1,

wherein said GPG program programmatically generating the new graphical program comprises the GPG program calling an application programming interface (API) enabling the programmatic generation of a graphical program.

15

25. The method of claim 1,

wherein said GPG program programmatically generating the new graphical program comprises the GPG program programmatically requesting a server program to generate the new graphical program.

20

26. The method of claim 25,

wherein the server program is an application instance of a graphical programming environment.

25

27. The method of claim 1,

wherein the GPG program comprises a client portion and a server portion;

wherein the client portion is operable to utilize an application programming interface (API) in order to direct the server program to programmatically generate the new graphical program.

34. The method of claim 32, wherein the new graphical program includes a block diagram, wherein the at least one graphical program object comprises a programmatic structure placed in the block diagram.

5

35. The method of claim 32, wherein the new graphical program includes a user interface panel, wherein the at least one graphical program object comprises a user interface object placed in the user interface panel.

10 36. The method of claim 35, wherein the user interface object is a user interface input object placed in the user interface panel for performing one or more of: viewing input to the new graphical program; or providing input to the new graphical program.

15 37. The method of claim 35, wherein the user interface object is a user interface output object placed in the user interface panel for viewing output of the new graphical program.

20 38. The method of claim 35, wherein the new graphical program also includes a block diagram, wherein the user interface object is a user interface input object placed in the user interface panel for performing one or more of: viewing input to the block diagram; or providing input to the new graphical program.

25 39. The method of claim 35, wherein the new graphical program also includes a block diagram, wherein the user interface object is a user interface output object placed in the user interface panel for viewing output from the block diagram.

40. The method of claim 1, wherein said programmatically generating the new graphical program comprises:

including a first graphical program object and a second graphical program object in the new graphical program; and

connecting the first graphical program object to the second graphical program object.

5

41. The method of claim 40, wherein said connecting the first graphical program object to the second graphical program object comprises connecting an input of the first graphical program object to an output of the second graphical program object.

10

42. The method of claim 1,

wherein the GPG program is a graphical program;

wherein the GPG program includes at least one object creation node for programmatically creating at least one graphical program object in the new graphical program;

15

wherein said generating the new graphical program comprises including the at least one graphical program object in the new graphical program.

43. The method of claim 42, wherein the GPG program further includes a property node, the method further comprising:

20

the property node getting or setting a property of the graphical program object in response to said executing the GPG program.

44. The method of claim 43, wherein the object creation node outputs a reference to the graphical program object;

25

wherein the property node receives as input the reference to the graphical program object;

wherein the property node gets or sets a property of the graphical program object specified by the reference to the graphical program object.

45. The method of claim 42, wherein the GPG program further includes an invoke node; the method further comprising:

the invoke node invoking a method on the graphical program object in response to said executing the GPG program.

46. The method of claim 45, wherein the object creation node outputs a reference to the graphical program object;

wherein the invoke node receives as input the reference to the graphical program object;

wherein the invoke node invokes a method on the graphical program object specified by the reference to the graphical program object.

47. The method of claim 42, further comprising:

configuring the object creation node of the GPG program;

wherein said configuring comprises specifying a graphical program object class for the object creation node;

wherein the at least one graphical program object included in the new graphical program is of the graphical program object class.

48. The method of claim 1,

wherein the GPG program is a graphical program;

wherein the GPG program includes a graphical program creation node for programmatically creating the new graphical program.

49. A computer-implemented method for programmatically modifying an existing graphical program, comprising:

executing a GPG program;

the GPG program receiving information during program execution, wherein the information specifies functionality to add to the existing graphical program;

the GPG program programmatically modifying the existing graphical program in order to implement the specified functionality, in response to receiving the information.

5

50. The method of claim 49,

wherein said modifying the existing graphical program comprises adding graphical code to the existing graphical program.

10

51. The method of claim 49, wherein said programmatically modifying the existing graphical program modifies the existing graphical program without any user input specifying the modification to the existing graphical program during said modifying.

15

52. The method of claim 49,

wherein said receiving information during program execution comprises receiving user input specifying desired functionality to add to the existing graphical program.

20

53. A computer-implemented method for programmatically generating a new graphical program, comprising:

providing information specifying functionality of the new graphical program;

executing a graphical program generation (GPG) program;

the GPG program programmatically generating the new graphical program using said information, wherein the new graphical program implements the specified functionality.

25

54. The method of claim 53, wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

5 55. The method of claim 53, wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the new graphical program.

10 56. A memory medium for programmatically generating a new graphical program, the memory medium comprising program instructions executable to:

receive information, wherein the information specifies functionality of the new graphical program;

programmatically generate the new graphical program in response to said
15 information specifying the functionality of the new graphical program, wherein the new graphical program implements the specified functionality.

57. The memory medium of claim 56, wherein said programmatically generating the new graphical program creates the new graphical program without any
20 user input specifying the new graphical program during said creating.

58. The memory medium of claim 56, wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the new graphical program.

25 59. The memory medium of claim 56, wherein the new graphical program comprises a block diagram portion comprising a plurality of interconnected nodes and a user interface portion;

wherein said programmatically generating the new graphical program includes generating the block diagram portion and the user interface portion.

5 60. A system for programmatically generating a new graphical program, the system comprising:

 a processor coupled to a memory, wherein the memory stores a graphical program generation (GPG) program;

 wherein the processor is operable to execute the GPG program in order to:

10 receive information specifying functionality of the new graphical program; and

 programmatically generate the new graphical program in response to said information specifying the functionality of the new graphical program, wherein the new graphical program implements the specified functionality.

15 61. The system of claim 60, wherein said programmatically generating the new graphical program creates the new graphical program without any user input specifying the new graphical program during said creating.

20 62. The system of claim 60, wherein the new graphical program comprises a plurality of interconnected nodes that visually indicate functionality of the new graphical program.

25 63. The system of claim 60, wherein the new graphical program comprises a block diagram portion comprising a plurality of interconnected nodes and a user interface portion;

 wherein said programmatically generating the new graphical program includes generating the block diagram portion and the user interface portion.

64. The system of claim 60, wherein said programmatically generating the new graphical program comprises:

creating a plurality of graphical program objects in the new graphical program;
and

5 interconnecting the plurality of graphical program objects in the new graphical program;

wherein the interconnected plurality of graphical program objects comprise at least a portion of the new graphical program.

10